PGPUB-DOCUMENT-NUMBER: 20030014598

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030014598 A1

TITLE: Software raid methods and apparatuses including server usage based write delegation

PUBLICATION-DATE: January 16, 2003

INVENTOR-INFORMATION:

NAME RULE-47	CITY	STATE	COUNTRY
Brown, William P.	Bellevue	WA	US

US-CL-CURRENT: 711/141, 711/114 , 711/152

ABSTRACT:

At least a first and a second server of a cluster of servers are equipped with complementary software RAID drivers and distributed lock managers to enable the first server to delegate to the second server, writing of a version of a unit of coherent data into a number of storage devices coupled to the server cluster. The drivers and lock managers are designed to enable the first server to determine the second server as an appropriate current synchronization server target, which determination includes consideration of the last synchronization server target. If the last synchronization server target is not the appropriate current synchronization server target, the second server is selected among the "eligible" servers of the cluster. The consideration/selection may be based on the usage state of the candidate server.

---------- KWIC ---------

Detail Description Paragraph - DETX (5):

[0018] In accordance with another aspect of the present invention, in performing a **delegated** write, the **delegated** server may obtain at least a shared read **lock** on the unit of coherent data and **validating** a timestamp of the version of the unit of coherent data to be written. The **delegated** server may also notify one or more other servers to cancel any scheduled write, the one or more other servers may have for their versions of the unit of coherent data.

Claims Text - CLTX (43):

42. An article of manufacture comprising: a storage medium; a distributed **lock** manager stored in the storage medium, designed to program a server to enable the server to facilitate obtaining of **locks** from a partition **lock** manager and **validating** timestamps of units of coherent data with the partition **lock** manager; and a software RAID driver stored in the storage medium, also designed to program the server, to facilitate RAID writing of coherent data into a plurality of storage devices to which the server and other servers are coupled, and reading of the coherent data, including performing **delegated** writes for other servers, wherein for the performance of a **delegated** write, the software RAID driver is designed to receive from a second server of the cluster, a replicated copy of a first version of a unit of coherent data, to be written into the plurality of storage devices on behalf of the second server, schedule the requested write, obtain through the distributed **lock** manager, at least a shared read **lock** on the unit of the coherent data, **validate** through the

distributed **lock** manager, a timestamp of the replicated copy, obtain a prior version of the unit of coherent data and its parity data, compute new parity data for the first version of the unit of coherent data, write the first version of the unit of coherent data and the computed new parity data into the plurality of storage devices, and update the partition **lock** manager with a new write timestamp.

Claims Text - CLTX (47):

46. A server comprising: a distributed **lock** manager to enable the server to facilitate obtaining of **locks** from a partition **lock** manager and **validating** timestamps of units of coherent data with the partition **lock** manager; and a software RAID driver operationally coupled to the distributed **lock** manager to facilitate RAID writing of coherent data into a plurality of storage devices to which the server and other servers are coupled, and reading of the coherent data, including performing **delegated** writes for other servers, wherein for the performance of a **delegated** write, the software RAID driver is designed to receive from a second server of the cluster, a replicated copy of a first version of a unit of coherent data, to be written into the plurality of storage devices on behalf of the second server, schedule the requested write, obtain through the distributed **lock** manager, at least a shared read **lock** on the unit of the coherent data, **validate** through the distributed **lock** manager, a timestamp of the replicated copy, obtain a prior version of the unit of coherent data and its parity data, compute new parity data for the first version of the unit of coherent data, write the first version of the unit of coherent data and the computed new parity data into the plurality of storage devices, and update the partition **lock** manager with a new write timestamp.

Claims Text - CLTX (49):

48. The server of claim 47, wherein the second software RAID driver and the second distributed **lock** manager are designed to perform the **delegated** write by obtaining, at least a shared read **lock** on the unit of the coherent data, **validating** a timestamp of the replicated copy, obtaining a prior version of the unit of coherent data and its parity data, computing new parity data for the first version of the unit of coherent data, writing the first version of the unit of coherent data and the computed new parity data into the plurality of storage devices, and updating the partition **lock** manager with a new write timestamp.

Claims Text - CLTX (52):

51. A cluster of servers comprising: a first server having a first software RAID driver and a first distributed **lock** manager operationally coupled to each other to **delegate** to a coupled second server, writing of a first version of a unit of coherent data into a plurality of storage devices coupled to the cluster of servers; and the second server, having a second software RAID driver and a second distributed **lock** manager operationally coupled to each other to perform the **delegated** write on behalf of the first server; wherein for the performance of the **delegated** write, the second software RAID driver and the second distributed **lock** are designed to receive from the first server, a replicated copy of the-first version of the unit of coherent data, schedule the requested write, obtain at least a shared read **lock** on the unit of the coherent data, **validate** a timestamp of the replicated copy, obtain a prior version of the unit of coherent data and its parity data, compute new parity data for the first version of the unit of coherent data, write the first version of the unit of coherent data and the computed new parity data into the plurality of storage devices, and update the partition **lock** manager with a new write timestamp.